Announcements

• HW3 problem 4c
Announcements

• HW3 problem 4c
Announcements

• HW3 problem 4c
Sequences and Recurrent Neural Networks

Machine Learning – CSE4546
Kevin Jamieson
University of Washington

November 30, 2017
Variable length sequences

Images are usually standardized to be the same size (e.g., 256x256x3)

Neural Network
Variable length sequences

Images are usually standardized to be the same size (e.g., 256x256x3)

Neural Network

But what if we wanted to do classification on country-of-origin for names?

Recurrent Neural Network
Variable length sequences

Recurrence Neural Network

Standard RNN

LSTM

Slide: http://colah.github.io/posts/2015-08-Understanding-LSTMs/
Basic Text/Document Processing

Machine Learning – CSE4546
Kevin Jamieson
University of Washington

November 30, 2017
TF*IDF

How to get a feature representation of each article?

1. For each document $d$ compute the proportion of times word $t$ occurs out of all words in $d$, i.e. term frequency

$$TF_{d,t}$$

2. For each word $t$ in your corpus, compute the proportion of documents out of $n$ that the word $t$ occurs, i.e., document frequency

$$DF_t$$

3. Compute score for word $t$ in document $d$ as

$$TF_{d,t} \log \left( \frac{1}{DF_t} \right)$$
BeerMapper - Under the Hood

Algorithm requires feature representations of the beers \( \{x_1, \ldots, x_n\} \subset \mathbb{R}^d \)

Two Hearted Ale - Input ~2500 natural language reviews

http://www.ratebeer.com/beer/two-hearted-ale/1502/2/1/

Reviews for each beer

Bag of Words weighted by TF*IDF

Get 100 nearest neighbors using cosine distance

Non-metric multidimensional scaling

Embedding in d dimensions
BeerMapper - Under the Hood

Algorithm requires feature representations of the beers \( \{x_1, \ldots, x_n\} \subset \mathbb{R}^d \)

**Two Hearted Ale - Weighted Bag of Words:**

- Reviews for each beer
- Bag of Words weighted by TF*IDF
- Get 100 nearest neighbors using cosine distance
- Non-metric multidimensional scaling
- Embedding in \( d \) dimensions
BeerMapper - Under the Hood

Algorithm requires feature representations of the beers \( \{x_1, \ldots, x_n\} \subset \mathbb{R}^d \)

<table>
<thead>
<tr>
<th>Reviews for each beer</th>
<th>Bag of Words weighted by TF*IDF</th>
<th>Get 100 nearest neighbors using cosine distance</th>
<th>Non-metric multidimensional scaling</th>
<th>Embedding in d dimensions</th>
</tr>
</thead>
</table>

Weighted count vector for the \( i \)th beer:

\[ z_i \in \mathbb{R}^{400,000} \]

Cosine distance:

\[ d(z_i, z_j) = 1 - \frac{z_i^T z_j}{||z_i|| \cdot ||z_j||} \]

**Two Hearted Ale - Nearest Neighbors:**
Bear Republic Racer 5
Avery IPA
Stone India Pale Ale & IPAs
Founders Centennial IPA
Smuttynose IPA
Anderson Valley Hop Ottin IPA
AleSmith IPA
BridgePort IPA
Boulder Beer Mojo IPA
Goose Island India Pale Ale
Great Divide Titan IPA
New Holland Mad Hatter Ale
Lagunitas India Pale Ale
Heavy Seas Loose Cannon Hop3
Sweetwater IPA
BeerMapper - Under the Hood

Algorithm requires feature representations of the beers \( \{x_1, \ldots, x_n\} \subset \mathbb{R}^d \)

Find an embedding \( \{x_1, \ldots, x_n\} \subset \mathbb{R}^d \) such that
\[
\|x_k - x_i\| < \|x_k - x_j\| \quad \text{whenever} \quad d(z_k, z_i) < d(z_k, z_j)
\]
for all 100-nearest neighbors. (10^7 constraints, 10^5 variables)

Solve with hinge loss and stochastic gradient descent. (20 minutes on my laptop) (\( d=2, \text{err}=6\% \)) (\( d=3, \text{err}=4\% \))

Could have also used local-linear-embedding, max-volume-unfolding, kernel-PCA, etc.
BeerMapper - Under the Hood

Algorithm requires feature representations of the beers \( \{x_1, \ldots, x_n\} \subset \mathbb{R}^d \)

Reviews for each beer

Bag of Words weighted by TF*IDF

Get 100 nearest neighbors using cosine distance

Non-metric multidimensional scaling

Embedding in \( d \) dimensions
BeerMapper - Under the Hood

Algorithm requires feature representations of the beers \( \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d \)

- Reviews for each beer
- Bag of Words weighted by TF*IDF
- Get 100 nearest neighbors using cosine distance
- Non-metric multidimensional scaling
- Sanity check: styles should cluster together and similar styles should be close.

Embedding in \( d \) dimensions
BeerMapper - Under the Hood

Algorithm requires feature representations of the beers \( \{x_1, \ldots, x_n\} \subset \mathbb{R}^d \)

- Reviews for each beer
- Bag of Words weighted by TF*IDF
- Get 100 nearest neighbors using cosine distance
- Non-metric multidimensional scaling
- Embedding in d dimensions

Sanity check: styles should cluster together and similar styles should be close.
Other document modeling

Matrix factorization:

1. Construct word x document matrix of counts
2. Compute non-negative matrix factorization
3. Use factorization to represent documents
4. Cluster documents into topics

Also see latent Dirichlet factorization (LDA)
Previous section presented methods to embed documents into a latent space.

Alternatively, we can embed words into a latent space.

This embedding came from directly querying for relationships.

*word2vec* is a popular unsupervised learning approach that just uses a text corpus (e.g. *nytimes.com*).
Word embeddings, word2vec

Source Text

The quick brown fox jumps over the lazy dog.

Training Samples

(the, quick)
(the, brown)
(quick, the)
(quick, brown)
(quick, fox)
(brown, the)
(brown, quick)
(brown, fox)
(brown, jumps)
(fox, quick)
(fox, brown)
(fox, jumps)
(fox, over)

slide: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/
Word embeddings, word2vec

Training neural network to predict co-occurring words. Use first layer weights as embedding, throw out output layer

slide: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/
Training neural network to predict co-occurring words. Use first layer weights as embedding, throw out output layer

slide: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/
word2vec outputs

king - man + woman = queen

country - capital

slide: https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/
Active Learning, classification

Machine Learning – CSE4546
Kevin Jamieson
University of Washington

November 30, 2017
Impressive recent advances in image recognition and translation…
Impressive recent advances in image recognition and translation…

Challenges for large models:

1) An enormous amount of **labeled data** is necessary for training
Impressive recent advances in image recognition and translation…

Challenges for large models:

1) An enormous amount of *labeled data* is necessary for training

2) An enormous amount of *wall-clock time* is necessary for training
Example: Image recognition

airplane
automobile
bird
Example: Image recognition

- airplane
- automobile
- bird
Example: Image recognition

Nonadaptive label assignment

- airplane
- automobile
- bird
Example: Image recognition

Nonadaptive label assignment

airplane
automobile
bird
Example: Image recognition

Nonadaptive label assignment

Adaptive label assignment

airplane
automobile
bird
Example: Image recognition

Nonadaptive label assignment

Adaptive label assignment

Examples:
- airplane
- automobile
- bird
$x_1, x_2, \ldots$ i.i.d.  
$x_j$ may depend on $\{x_i\}_{i<j}$

complexity (reliability/robustness, scalability/computation, etc)
Being convinced that data-collection *should be adaptive* is not the same thing as knowing *how to be adaptive*. 
Third  “Maybe his second week will go better”
Second “I’d like to see other people”
First  “The corrupt media will blow this way out of proportion”
• $n \approx 5000$ captions submitted each week
• crowdsourcing contest to volunteers who rate captions
• goal: identify funniest caption

newyorker.com/cartoons/vote
“It's amazing to think he started out in the lobby.”
"I thought all our plants moved to Mexico."
“Be patient. He'll grow on you.”

Which caption do we show next?

1) **Non-adaptive** uniform distribution over captions
2) **Adaptive:** stop showing captions that will not win
4-5 times fewer ratings needed

Which caption do we show next?

1) **Non-adaptive** uniform distribution over captions
2) **Adaptive**: stop showing captions that will not win
**Objective**: with probability .99, identify \( \arg\max_{i=1,\ldots,n} \mu_i \) using as few total samples as possible.

While the algorithm does not exit:
- The algorithm shows caption \( i \in \{1,\ldots,n\} \)
- Observe iid Bernoulli with \( P(\text{“funny”}) = \mu_i \)

**Stopping rule**

**Sampling rule**
Best-arm Identification \( n = 2 \)

Consider \( n = 2 \) and flip coins \( i = 1, 2 \) to get \( X_{i,1}, X_{i,2}, \ldots, X_{i,m} \)

\[
\begin{align*}
\hat{\mu}_{i,m} &= \frac{1}{m} \sum_{j=1}^{m} X_{i,j} \\
\text{Test: } \hat{\mu}_{1,m} - \hat{\mu}_{2,m} &\geq 0
\end{align*}
\]

By a Chernoff Bound, if \( \Delta = \mu_1 - \mu_2 \) then

\[
m = 2 \log(1/\delta) \Delta^{-2} \implies \hat{\mu}_{1,m} > \hat{\mu}_{2,m} + 2 \sqrt{\frac{\log(1/\delta)}{2m}} \implies \mu_1 > \mu_2
\]

with probability \( \geq 1 - 2\delta \)

Arm 1 lower confidence bound > Arm 2 upper confidence bound
confidence interval $\propto \sqrt{\frac{\log(n)}{\#\text{votes}}}$

average $\#\text{ votes}$ for “Funny”
confidence interval \( \propto \sqrt{\frac{\log(n)}{\# \text{votes}}} \)

more votes \( \implies \) smaller intervals
keep sampling until intervals do not overlap

confidence interval $\propto \sqrt{\frac{\log(n)}{\#\text{votes}}}$
# votes Non-adaptive: \( n \max_{i=1, \ldots, n} \Delta_i^{-2} \log(n) \)

Successive Elimination [Even-dar…’06]: \( \sum_{i=1}^{n} \Delta_i^{-2} \log(n) \)

Stop sampling caption \( i \) as soon as no overlap

\[
\text{confidence interval } \propto \sqrt{\frac{\log(n)}{\text{#votes}}}
\]

keep sampling until intervals do not overlap

\( \{ \Delta_2, \Delta_3 \} \)
Learn an accurate classifier using a small number of labels

Find the winner of a competition using a small number of judgements
Very related to adaptive A/B testing

Find the ad that results in highest click-through-rate and keep showing it

Balance of exploration versus exploitation